

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (previously presented) A process for monitoring, comprising:
accessing a method;
determining whether to modify said method, said step of determining whether to modify said method includes determining whether said method calls another method; and
modifying said method for a particular purpose if said method calls another method.
2. (currently amended) The process according to claim 1, wherein:
said step of determining whether to modify said method includes determining whether said method is a [[non-]]synthetic method; and
said step of modifying includes modifying said method if said method is not a [[non-]]synthetic method and said method calls another method.
3. (currently amended) The process according to claim 1, wherein:
said step of determining whether to modify said method includes determining ~~whether said method has an access level of said method of public or package~~; and
said step of modifying includes modifying said method if said method has [[said]] an access level that satisfies a criterion ~~of public or package~~ and said method calls another method.
4. (cancelled)
5. (currently amended) The process according to claim 1, wherein:
said step of determining whether to modify said method includes determining whether said method is a [[non-]]synthetic method and has an access level of public or package in the JAVA programming language; and

said step of modifying includes modifying said method if said method is not a [[non-
]]synthetic method, has said access level of public or package, and said method calls another method.

6. (previously presented) The process according to claim 1, wherein:
said step of determining whether to modify said method includes determining whether said method can be called by a sufficient scope of one or more other methods, ~~said method has an access level~~, said determining whether said method can be called by a sufficient scope of one or more other methods is based on [[said]] an access level of said method; and
said step of modifying said method includes modifying said method if said method can be called by said sufficient scope of one or more other methods and said method calls another method.

7. (previously presented) The process according to claim 1, wherein:
said step of modifying includes modifying object code.

8. (previously presented) The process according to claim 1, wherein:
said step of modifying includes adding a tracer for said method.

9. (previously presented) The process according to claim 1, wherein:
said step of modifying includes adding a timer for said method.

10. (previously presented) The process according to claim 1, wherein:
said step of modifying includes adding exit code and start code to existing object code.

11. (previously presented) The process according to claim 10, wherein:
said start code starts a tracing process;
said exit code stops said tracing process;
said exit code is positioned to be executed subsequent to original object code;
said step of adding exit code includes adding an instruction to jump to said exit code from

said original object code;

said step of adding exit code includes adding an entry in an exceptions table; and

said step of adding an entry in said exceptions table includes adding a new entry into said exceptions table for said method, said new entry indicates a range of indices corresponding to said original object code, said new entry includes a reference to said exit code and said new entry indicates that said new entry pertains to all types of exceptions.

12. (previously presented) The process according to claim 1, wherein:
said particular purpose is to add a first tracer.

13. (currently amended) A process for monitoring, comprising:
determining which methods of a set of methods satisfy criteria for likely being at the top of a call graph without using information from a call graph ~~call one or more other methods~~; and
using a first tracing mechanism for said methods that satisfy the criteria ~~determined to call one or more other methods~~ without using said first tracing mechanism for methods that do not satisfy the criteria ~~determined to call one or more other methods~~.

14. (currently amended) The process according to claim 13, wherein:
said step of determining includes determining whether said methods are [[non-]]synthetic methods; and
said step of using includes using said first tracing mechanism if said methods are determined to not be [[non-]]synthetic methods and said methods call one or more other methods.

15. (currently amended) The process according to claim 13, wherein:
said step of determining includes determining whether said methods have an access level of public or package in the JAVA programming language; and
said step of using includes using said first tracing mechanism if said methods are determined to have said access level of public or package and said methods call one or more other methods.

16. (cancelled)

17. (currently amended) The process according to claim 13, wherein:

said step of determining includes determining whether said methods are [[non-]]synthetic methods and have an access level of public or package in the JAVA programming language; and

said step of using includes using said first tracing mechanism if said methods are not [[non-]]synthetic methods, have said access level of public or package, and said methods call one or more other methods.

18. (previously presented) The process according to claim 13, wherein:

said step of determining includes determining whether said methods can be called by a sufficient scope of one or more other methods; and

said step of using includes using said first tracing mechanism if said methods can be called by said sufficient scope of one or more other methods and said methods call one or more other methods.

19. (previously presented) The process according to claim 13, wherein:

said step of using a first tracing mechanism includes adding and using timers for said methods.

20. (previously presented) The process according to claim 13, wherein:

said step of using a first tracing mechanism includes modifying existing object code to add said first tracing mechanism.

21. (previously presented) The process according to claim 20, wherein:

said first tracing mechanism includes timers for said methods.

22. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor

readable code for programming one or more processors to perform a process comprising:

determining which methods of a set of methods to modify, said step of determining includes determining which methods satisfy a set of one or more criterion for likely being at the top of a call graph;

~~whether said methods call one or more other methods~~; and

modifying for a particular purpose only those methods that satisfy the set of one or more criterion ~~are determined to call one or more other methods~~.

23. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said step of determining includes determining whether said methods are not ~~[[non-]]~~synthetic methods; and

said step of modifying includes modifying said methods if said methods are determined to not be ~~[[non-]]~~synthetic methods and said methods call one or more other methods.

24. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said step of determining includes determining whether said methods have an access level of public or package in the JAVA programming language; and

said step of modifying includes modifying said methods determined to have said access level of public or package and said methods call one or more other methods.

25. (cancelled)

26. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said step of determining includes determining whether said methods are not ~~[[non-]]~~synthetic methods and have an access level of public or package in the JAVA programming language; and

said step of modifying includes modifying said methods if said methods are determined to

not be [[non-]]synthetic methods, have said access level of public or package, and said methods call one or more other methods.

27. (previously presented) The one or more processor readable storage devices according to claim 22, wherein:

said step of determining includes determining whether said methods can be called by a sufficient scope of one or more other methods; and

said step of modifying includes modifying said methods if said methods can be called by said sufficient scope of one or more other methods and said methods call one or more other methods.

28. (previously presented) The one or more processor readable storage devices according to claim 22, wherein:

said step of modifying includes modifying existing object code.

29. (previously presented) The one or more processor readable storage devices according to claim 22, wherein:

said step of modifying includes adding tracers.

30. (previously presented) The one or more processor readable storage devices according to claim 22, wherein:

said step of modifying includes adding timers.

31. (cancelled)

32. (currently amended) The one or more processor readable storage devices according to claim [[31]] 22, wherein:

said start code starts a tracing process;

said exit code stops said tracing process;

said exit code is positioned to be executed subsequent to original object code;

said step of adding exit code includes adding an instruction to jump to said exit code from said original object code;

said step of adding exit code includes adding an entry in an exceptions table; and

said step of adding an entry in said exceptions table includes adding a new entry into said exceptions table for said method, said new entry indicates a range of indices corresponding to said original object code, said new entry includes a reference to said exit code and said new entry indicates that said new entry pertains to all types of exceptions.

33. (previously presented) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a process comprising:

determining whether to trace a method, said step of determining includes determining whether said method calls another method; and

tracing said method for a particular purpose if said method calls another method.

34. (previously presented) The one or more processor readable storage devices according to claim 33, wherein:

said step of determining includes determining whether or not said method is flagged by a compiler as being synthetic; and

said step of tracing includes tracing said method if said method is not flagged by said compiler as being synthetic and said method calls another method.

35. (currently amended) The one or more processor readable storage devices according to claim 33, wherein:

said step of determining includes determining whether said method has an access level of public or package in the JAVA programming language, an access level of public indicates that a method can be called by a method in a class of any parentage, an access level of package indicates that a method can be called by methods in classes in the same package regardless of parentage; and

said step of tracing includes tracing said method if said method is determined to have said

access level of public or package and said method calls another method.

36. (cancelled)

37. (currently amended) The one or more processor readable storage devices according to claim 33, wherein:

said step of determining includes determining whether said method is not a [[non-]]synthetic method and has an access level of public or package, said access level is one of a plurality of access levels in a JAVA programming language; and

said step of tracing includes tracing said method if said method is determined to not be a [[non-]]synthetic method, have said access level of public or package, and said method calls another method.

38. (previously presented) The one or more processor readable storage devices according to claim 33, wherein:

said step of tracing includes timing said method.

39. (currently amended) An apparatus capable of monitoring, comprising:
means for determining whether a method calls another method;
means for determining whether said method can be called by a sufficient scope of one or more other methods;

means for determining whether said method is not a synthetic method; and
means for tracing said method for a particular purpose only if said method calls another method, said method can be called by a sufficient scope of one or more other methods, and said method is not a synthetic method.

40. (previously presented) An apparatus capable of monitoring, comprising:
a storage device; and
one or more processors in communication with said storage device, said one or more

processors perform a process comprising:

accessing a method,
determining whether said method calls one or more different methods and can be called by a sufficient scope of one or more other methods, and
tracing said method for a particular purpose if said method calls one or more different methods and can be called by a sufficient scope of one or more other methods.

41. (currently amended) The apparatus according to claim 40, wherein:
said step of determining includes determining whether said method is not a [[non-]]synthetic method; and

said step of tracing includes tracing said method if said method is determined to not be a [[non-]]synthetic method and said method calls one or more different methods.

42. (currently amended) The apparatus according to claim 40, wherein:
said step of determining includes determining whether said method has an access level of public or package in the JAVA programming language; and

said step of tracing includes tracing said method if said method is determined to have said access level of public or package and said method calls one or more different methods.

43. (cancelled)

44. (previously presented) The apparatus according to claim 40, wherein:
said process further includes modifying existing object code for said method in order to add a first tracing mechanism.

45. (previously presented) The apparatus according to claim 44, wherein:
said first tracing mechanism includes a timer.

46. (previously presented) The apparatus according to claim 40, wherein:

said step of tracing includes timing said method.

47. (currently amended) A process for monitoring, comprising:
accessing a method;
determining whether said method is complex, said step of determining includes determining that said method is complex if said method satisfies at least one of the following criteria:
said method calls another method;
said method has an access level that satisfies a criterion ~~is either public or package~~; or
said method is not flagged by a compiler as being synthetic; and
adding a tracer to said method only if said method is determined to be complex.

48. (previously presented) The process according to claim 47, wherein:
said step of determining includes determining that said method is complex if said method is not flagged by said compiler as being synthetic and said method calls another method.

49. (currently amended) The process according to claim 47, wherein:
said step of determining includes determining that said method is complex if said method has an access level that satisfies the criterion ~~of public or package~~, said method is not flagged by said compiler as being synthetic, and said method calls another method.

50. (cancelled)

51. (previously presented) The process according to claim 5, wherein:
said step of modifying includes adding a tracer for said method.

52. (currently amended) The apparatus according to claim 40, wherein:
said step of determining includes determining whether said method is not a ~~[[non-]]~~synthetic method and whether said method has an access level of public or package in the JAVA programming language; and

said step of tracing includes tracing said method if said method is determined to not be a [[non-]]synthetic method, said method is determined to have an access level of public or package, and said method calls one or more different methods.

53. (previously presented) The process of claim 47, wherein said step of determining includes determining that said method is complex if said method satisfies at least two of said criteria.

54. (previously presented) The process of claim 47, wherein said step of determining includes determining that said method is complex if said method calls another method.

55. (New) A method, comprising:
dividing a plurality of methods into a first group and a second group, each method is placed into either the first group or the second group based on whether the method calls another method, whether the method is synthetic, and an access level of the method; and
adding monitoring code to each method in the first group but not to any methods in the second group.

56. (New) The method of claim 55, wherein the dividing a plurality of methods into a first group and a second group includes placing methods that call another method into the first group.

57. (New) The method of claim 56, wherein the dividing a plurality of methods into a first group and a second group further includes placing methods that are synthetic into the second group.

58. (New) The method of claim 57, wherein the dividing a plurality of methods into a first group and a second group further includes placing methods that have an access level of either public or private in the JAVA programming language into the second group.

59. (New) The one or more processor readable storage devices according to claim 22, wherein said determining which methods satisfy a set of one or more criterion for likely being at the top of a call graph is not based on a call graph.